CLAIMS

What is claimed is:

1. A computer system, comprising:

a host processor, and

a network interface coupled to the host processor and to a network, the network interface comprising:

a first port that receives data from the host processor;

a second port that transmits data to the network;

a memory that stores data packets received by the first port, the memory being coupled to the first port and to the second port;

and

a control circuit that manages the memory as a plurality of queues having respective priorities, including logic to place a packet received from the host into one of the plurality of queues according to a quality of service parameter associated with the packet, and logic to service packets in the plurality of queues according to the respective priorities.

2. The computer system of claim 1, wherein the plurality of queues includes a higher priority queue, and a lower priority queue, and including a timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a timeout interval, and including logic to preempt the higher priority queue in favor of the lower priority queue if the timeout timer expires.

3. The computer system of claim 1, wherein the plurality of queues includes a higher priority queue, an intermediate priority queue, and a lower priority queue, and including

a first timeout timer coupled with the intermediate priority queue which is enabled if a packet is stored in the intermediate priority queue and expires after a first timeout interval, and including logic to preempt the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires; and

a second timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a second timeout interval, and including

logic to preempt the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires.

4. The computer system of claim 1, wherein the plurality of queues includes a higher priority queue, an intermediate priority queue, and a lower priority queue, and including

an first timeout timer coupled with the intermediate priority queue which is enabled if a packet is stored in the intermediate priority queue and expires after a first timeout interval, and including logic to preempt the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires;

a second timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a second timeout interval, and including logic to preempt the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires; and

logic to service the intermediate priority queue in favor of the lower priority queue if both the first and second timeout timers expire.

5. The computer system of claim 1, further comprising logic in the network interface to execute a security process on packets in one of the plurality of queues.

6. The computer system of claim 1, wherein the second port further comprises circuitry for formatting packets according to a protocol compliant with an Ethernet protocol standard.

7. The computer system of claim 1, wherein the second port further comprises circuitry for formatting packets according to a protocol compliant with an Infiniband protocol standard.

8. The computer system of claim 1, wherein the packets include frame start headers, and said quality of service parameters comprises codes in the frame start headers.

9. The computer system of claim 1, wherein said plurality of queues have statically allocated space in said memory.

10. The computer system of claim 1, wherein said plurality of queues include first, second and third queues, and said memory includes a first storage array for the first queue, a second storage array for the second queue, and a third storage array for the third queue, and wherein said first, second and third storage arrays have respective inputs coupled to said logic to place a packet in one of the plurality of queues, and respective outputs, and wherein said logic to service packets in the plurality of queues includes a multiplexer coupled to the outputs of the first, second and third storage arrays.

11. The computer system of claim 1, wherein at least one of the queues in the plurality of queues comprises a first-in-first-out FIFO queue.

12. In a network interface apparatus, a method for managing transfer of data packets between a host processor and a network, comprising:

managing memory in the network interface apparatus as a plurality of queues having respective priorities, including placing a packet received from the host processor into one of the plurality of queues according to a quality of service parameter associated with the packet, and servicing packets in the plurality of queues according to the respective priorities.

13. The method of claim 12, wherein the plurality of queues includes a higher priority queue, and a lower priority queue, and including enabling a timeout timer coupled with the lower priority queue if a packet is stored in the lower priority queue, the timeout timer expiring after a timeout interval, and preempting the higher priority queue in favor of the lower priority queue if the timeout timer expires.

14. The method of claim 12, wherein the plurality of queues includes a higher priority queue, an intermediate priority queue, and a lower priority queue, and including

enabling a first timeout timer coupled with the intermediate priority queue if a packet is stored in the intermediate priority queue, the first timeout timer expiring after a first timeout interval, and preempting the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires; and

enabling a second timeout timer coupled with the lower priority queue if a packet is stored in the lower priority queue, the second timeout timer expiring after a second timeout

interval, and preempting the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires.

15. The method of claim 12, wherein the plurality of queues includes a high priority queue, an intermediate priority queue, and a lower priority queue, and including

enabling a first timeout timer coupled with the intermediate priority queue if a packet is stored in the intermediate priority queue, the first timeout timer expiring after a first timeout interval, and preempting the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires; and

enabling a second timeout timer coupled with the lower priority queue if a packet is stored in the lower priority queue, the second timeout timer expiring after a second timeout interval, and preempting the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires; and

servicing the intermediate priority queue in favor of the lower priority queue if both the first and second timeout timers expire.

16. The method of claim 12, further comprising executing a security process on packets in one of the plurality of queues.

17. The method of claim 12, including formatting packets in the network interface device according to a protocol compliant with an Ethernet protocol standard.

18. The method of claim 12, including formatting packets in the network interface device packets according to a protocol compliant with an Infiniband protocol standard.

19. The method of claim 12, wherein the packets include frame start headers, and said quality of service parameters comprises codes in the frame start headers.

20. The method of claim 12, wherein said plurality of queues have statically allocated space in said memory.

21. The method of claim 12, wherein said plurality of queues include first, second and third queues, and said network interface device includes memory comprising a first storage array for the first queue, a second storage array for the second queue, and a third storage array for the third queue, and including servicing packets in the plurality of queues using a multiplexer in the network interface device coupled to the outputs of the first, second and third storage arrays.

22. The method of claim 12, including managing at least one of the queues in said plurality of queues as a FIFO queue.

23. An integrated circuit for use in a network interface between a host processor and a network, comprising:

a first port that receives data from the host processor;

a second port that transmits data to the network;

a memory that stores data packets received by the first port, the memory being coupled to the first port and to the second port; and

a control circuit that manages the memory as a plurality of queues having respective priorities, including logic to place a packet received on the first port into one of the plurality of queues according to a quality of service parameter associated with the packet, and logic to transmit packets in the plurality of queues out the second port according to the respective priorities.

24. The integrated circuit of claim 23, wherein the plurality of queues includes a high priority queue, and a lower priority queue, and including a timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a timeout interval, and including logic to preempt the higher priority queue in favor of the lower priority queue if the timeout timer expires.

25. The integrated circuit of claim 23, wherein the plurality of queues includes a high priority queue, an intermediate priority queue, and a lower priority queue, and including

a first timeout timer coupled with the intermediate priority queue which is enabled if a packet is stored in the intermediate priority queue and expires after a first timeout interval, and

including logic to preempt the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires; and

a second timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a second timeout interval, and including logic to preempt the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires.

26. The integrated circuit of claim 23, wherein the plurality of queues includes a high priority queue, an intermediate priority queue, and a lower priority queue, and including

an first timeout timer coupled with the intermediate priority queue which is enabled if a packet is stored in the intermediate priority queue and expires after a first timeout interval, and including logic to preempt the higher priority queue in favor of the intermediate priority queue if the first timeout timer expires;

a second timeout timer coupled with the lower priority queue which is enabled if a packet is stored in the lower priority queue and expires after a second timeout interval, and including logic to preempt the higher priority queue and the intermediate priority queue in favor of the lower priority queue if the second timeout timer expires; and

logic to service the intermediate priority queue in favor of the lower priority queue if both the first and second timeout timers expire.

27. The integrated circuit of claim 23, further comprising logic in the network interface to execute a security process on packets in one of the plurality of queues.

28. The integrated circuit of claim 23, wherein the second port further comprises media access control circuitry for transmitting packets according to a protocol compliant with an Ethernet protocol standard.

29. The integrated circuit of claim 23, wherein the second port further comprises media access control circuitry for transmitting packets according to a protocol compliant with an Infiniband protocol standard.

30. The integrated circuit of claim 23, wherein the packets include frame start headers, and said quality of service parameters comprises codes in the frame start headers.

31. The integrated circuit of claim 23, wherein said plurality of queues have statically allocated space in said memory.

32. The integrated circuit of claim 23, wherein said plurality of queues include first, second and third queues, and said memory includes a first storage array for the first queue, a second storage array for the second queue, and a third storage array for the third queue, and wherein said first, second and third storage arrays have respective inputs coupled to said logic to place a packet in one of the plurality of queues, and respective outputs, and wherein said logic to service packets in the plurality of queues includes a multiplexer coupled to the outputs of the first, second and third storage arrays.

33. The integrated circuit of claim 23, at least one of the queues in the plurality of queues comprises a FIFO queue.